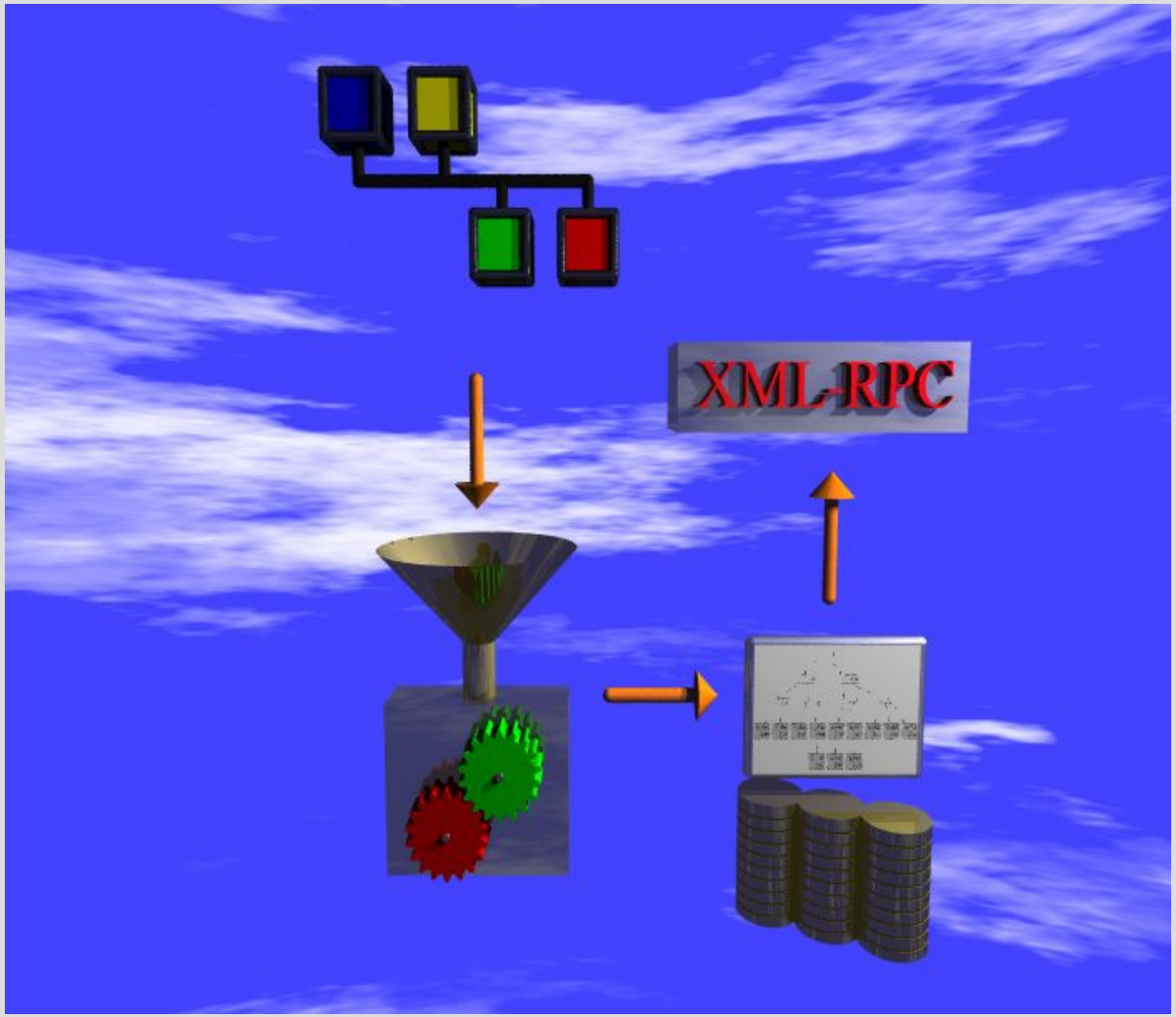
Channel Archiver Update

kasemir@lanl.gov



Issues

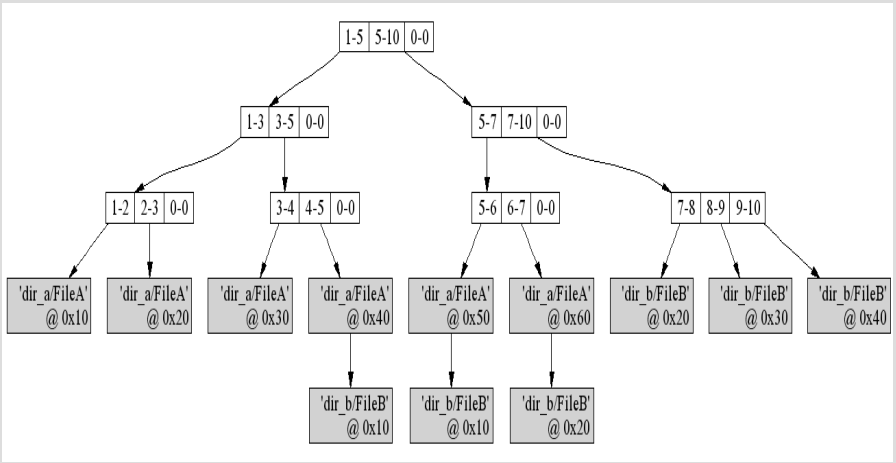
- Needed to support EPICS Base R3.14.
- Archive Engine's scanning code was a mess.
- Data Retrieval degraded poorly with amount of data.
- Plethora of retrieval tools, all bad. No portable and reasonable GUI retrieval tool.
- Binding to tcl, perl, python was hard to maintain, yet missing Java, Matlab.
- No network data server, and the one we didn't have lacked server-side data reduction.

R3.14 Port

- Using R3.14 eliminated own compatibility layer for threads, semaphores, network.
- Minimized use of C++ std:: library to list and vector.
  - Gone: namespaces, exceptions, iostream.
- Substantial Rewrite of ArchiveEngine
  - Using Multithreaded CA client lib.
  - Multithreaded HTTPD.
  - XML configuration file.
  - Tested under 'valgrind' memory checker.

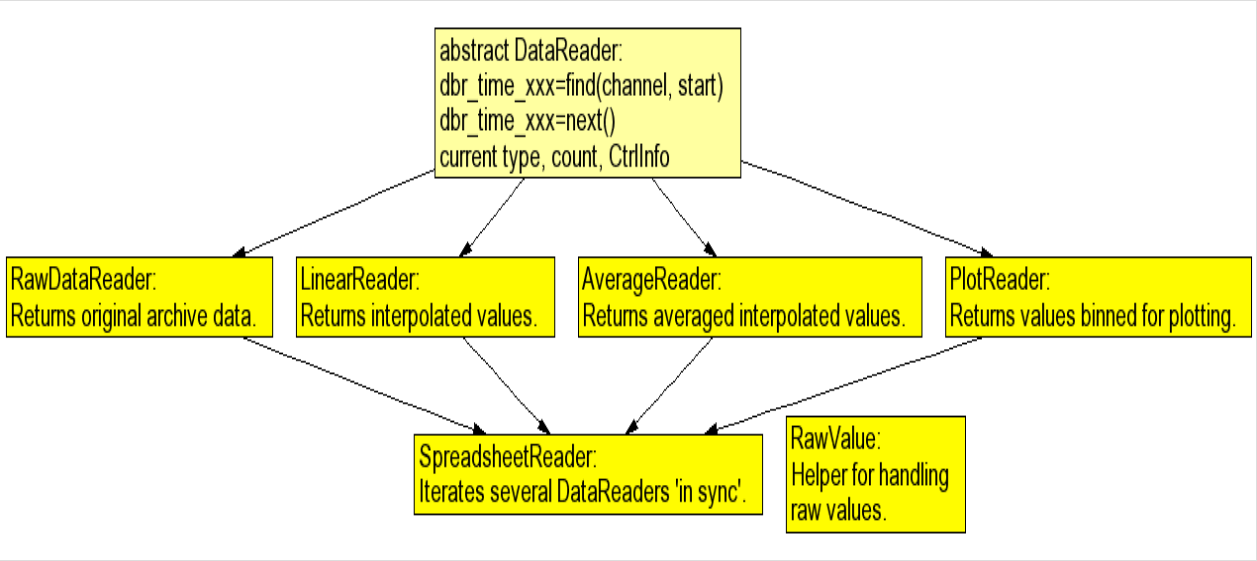
New Index

- Replaces old “Directory Files”
  - Pointer to first and last data block per channel.
  - Accessed collection of archives one-by-one.
- New Index: RTree-based
  - Keeps track of all data blocks.
    (Details to be presented by Sergei Chevtsov).
  - Allows creation of index from several sub-indices.



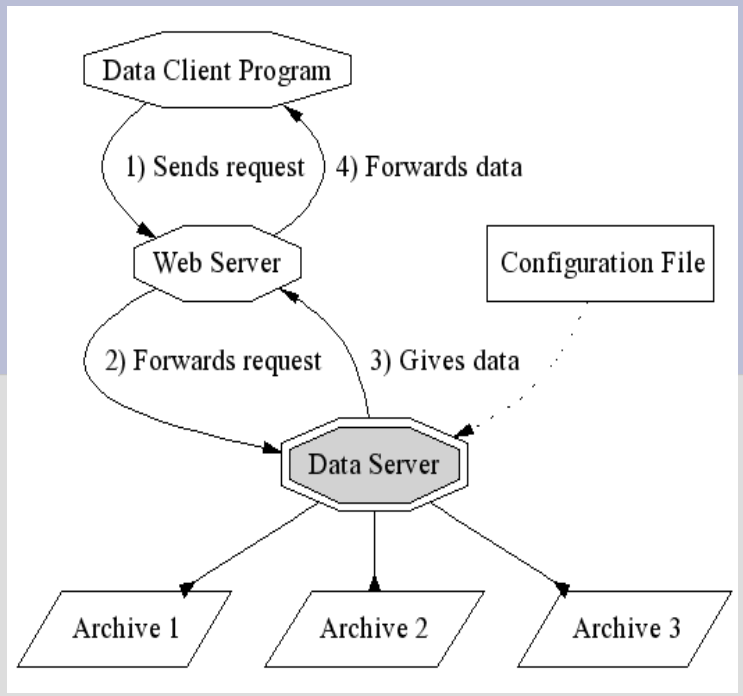
Data Access

- New C++ “Storage” library replaces “LibIO”
  - Concentrates on reading ChannelArchiver Data
  - No attempt to support extensions that never happen
  - More “Binning”



- “But Wait, there's more”

Network Data Server



- XML-RPC
  - Runs under web server, using its security mechanisms.
  - HTTP, passes through common firewalls.
  - Can serve multiple archives (config: XML file).
  - The XML-RPC-based protocol is now THE data access API. (replacing LibIO & glue code)
  - Natively accessible from C/C++, perl, Java

Network Data Server...

- XML-RPC
  - Transfers all data in XML, i.e. ASCII Text.
  - 20 byte dbr_time_double turns into >300 bytes XML.
  - So far performance is acceptable, especially considering universal accessibility of XML-RPC.
  - Data Server imposes limit of 10000 vals/channel, clients will then have to send another query.

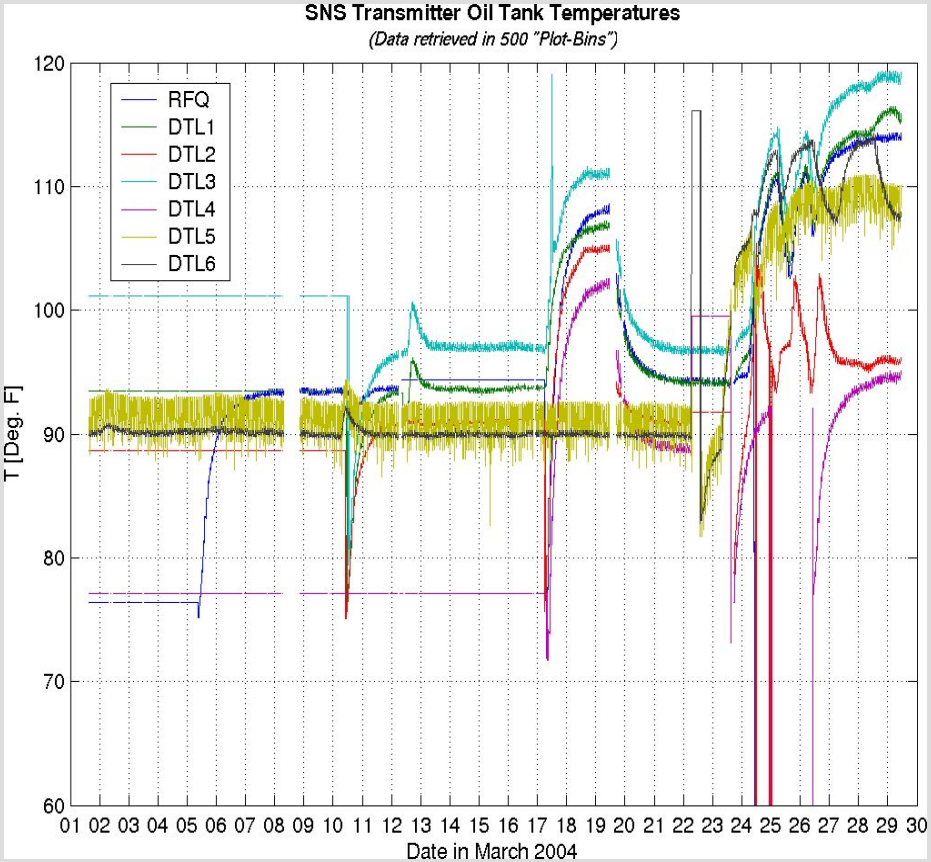
Java Client

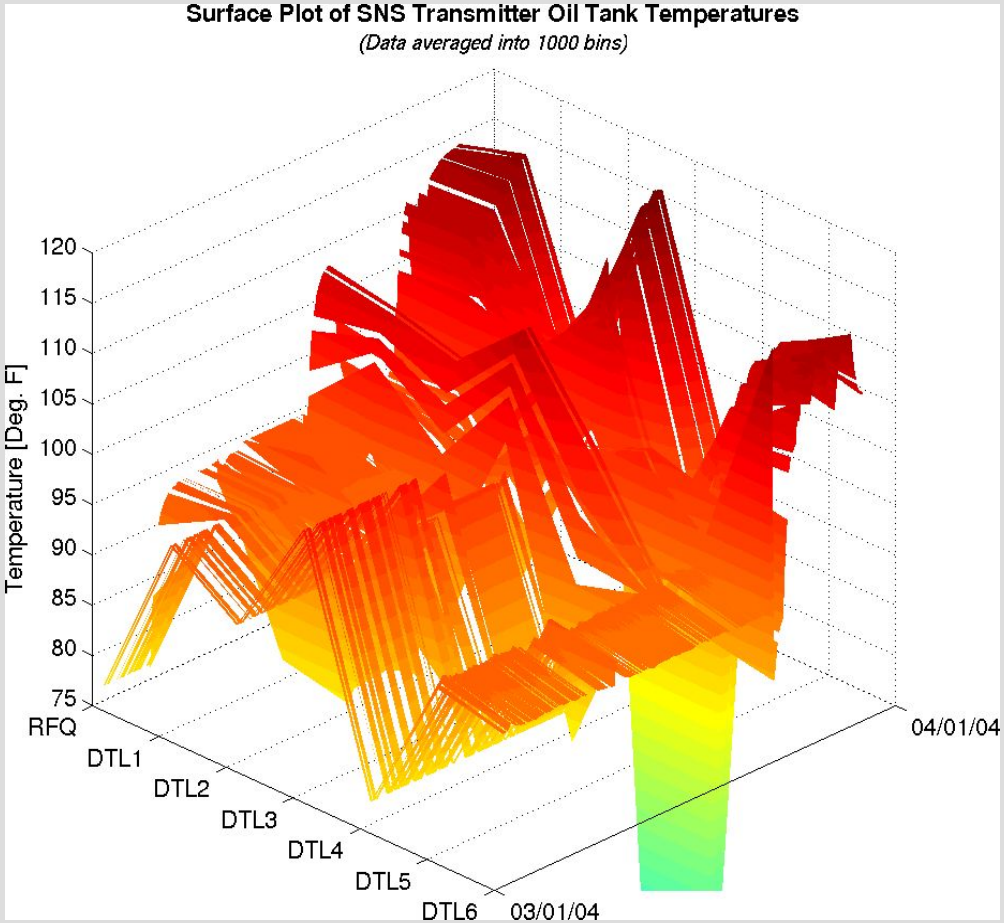
- Uses the Network Data Server
  (Details to be presented by Craig McChesney)

- Plot-Binning Example
  (ala Thomas Birke's CGIExport idea):
  7 ch,
  each ~23000 vals.
  reduced to
  800 bins
  (~1500 vals/ch).

Matlab/Octave

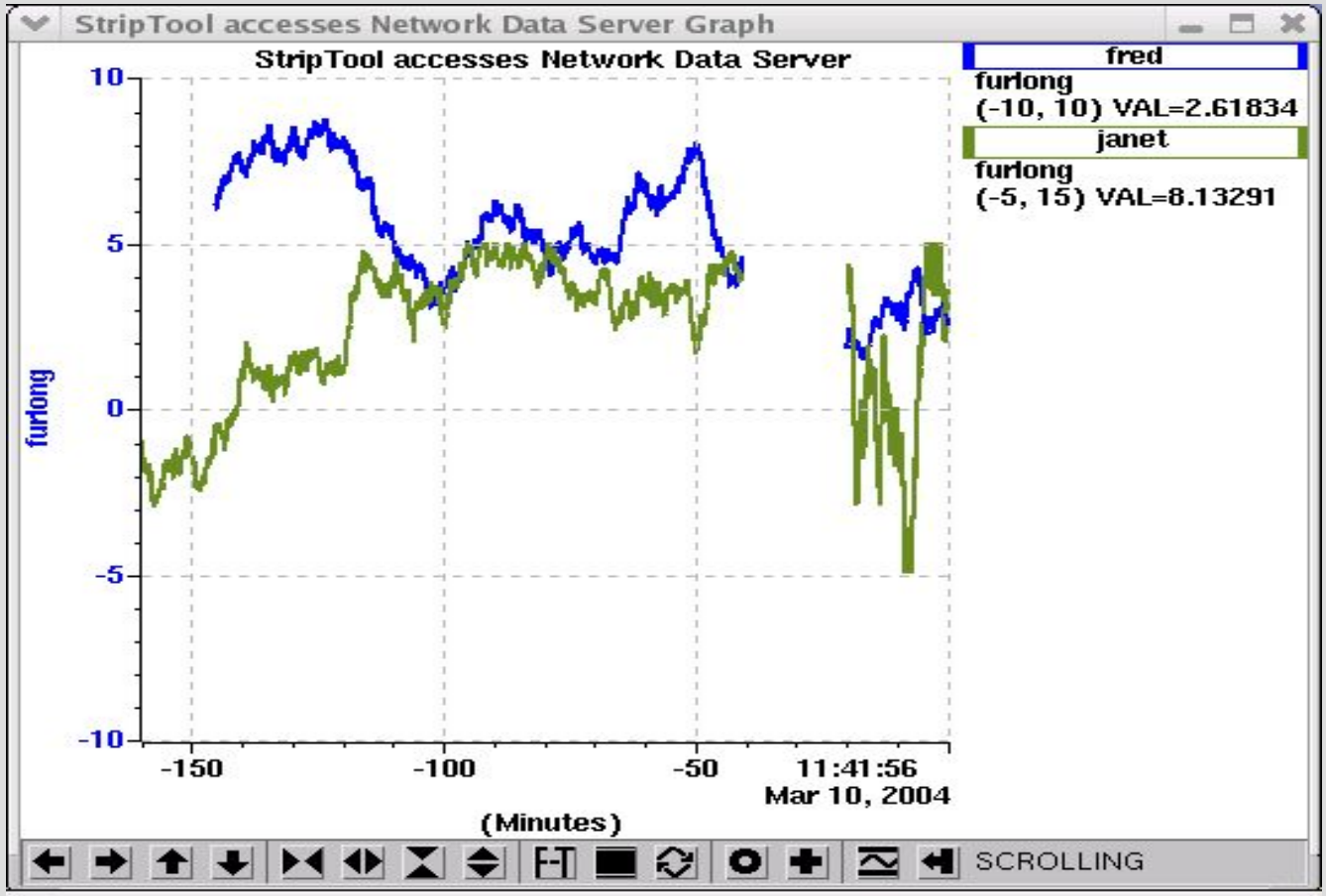
- MEX/OCT Interface to Network Data Server





StripTool

- Accesses Network Data Server.



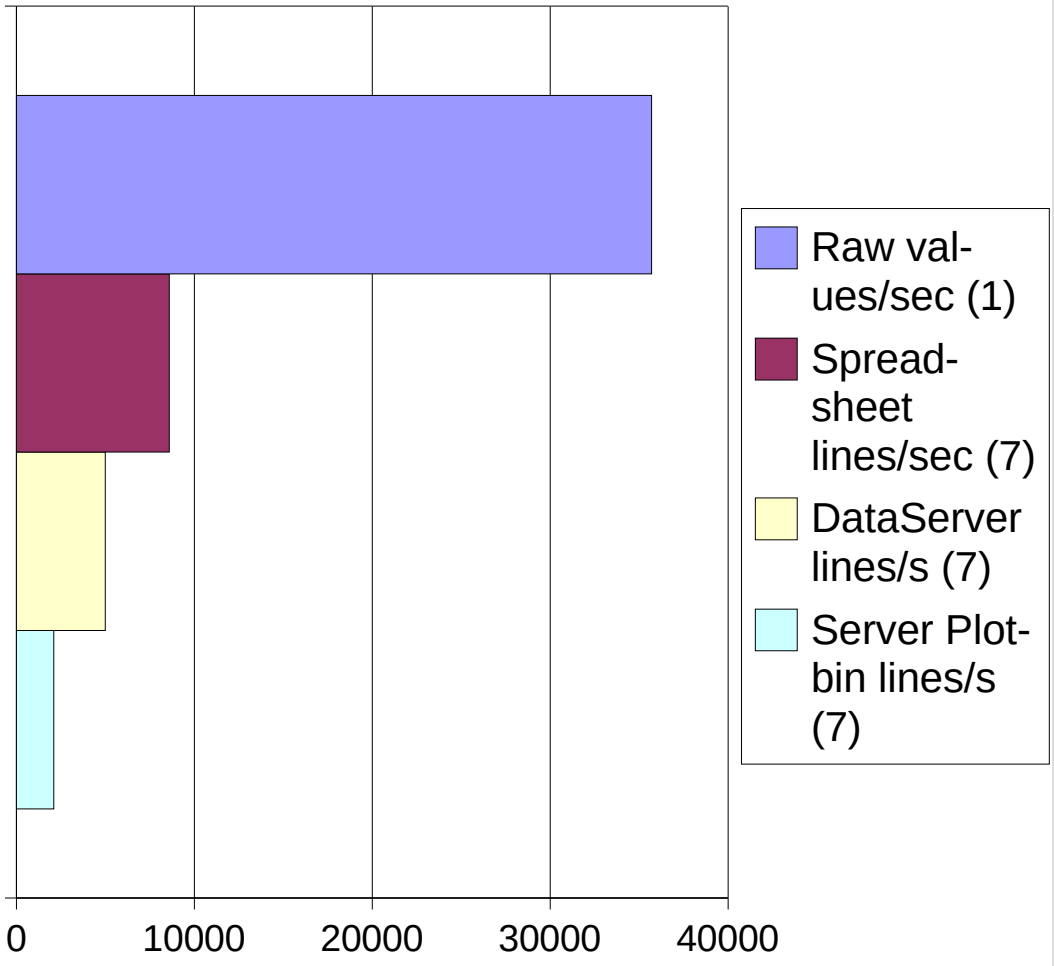
Performance

...will differ in your case...

Your mileage will vary. No warrantee is expressed or implied. Slippery when wet.

- Tests:
  - Disk performance outweighs CPU speed.
  - Reproducibility impacted by disk cashing.
  - Data management (fewer files) pays.
- Index:
  - Create master index for 12 sub-archives
    (one year, 1.4GB of LANL Xmtr test data): <1min
  - List/search master index for ~600 small sub-archs:
    <1 sec, compared to up to 40 secs w/ previous “Multi” list.
    (Creation time for that master of >600: 15 minutes)

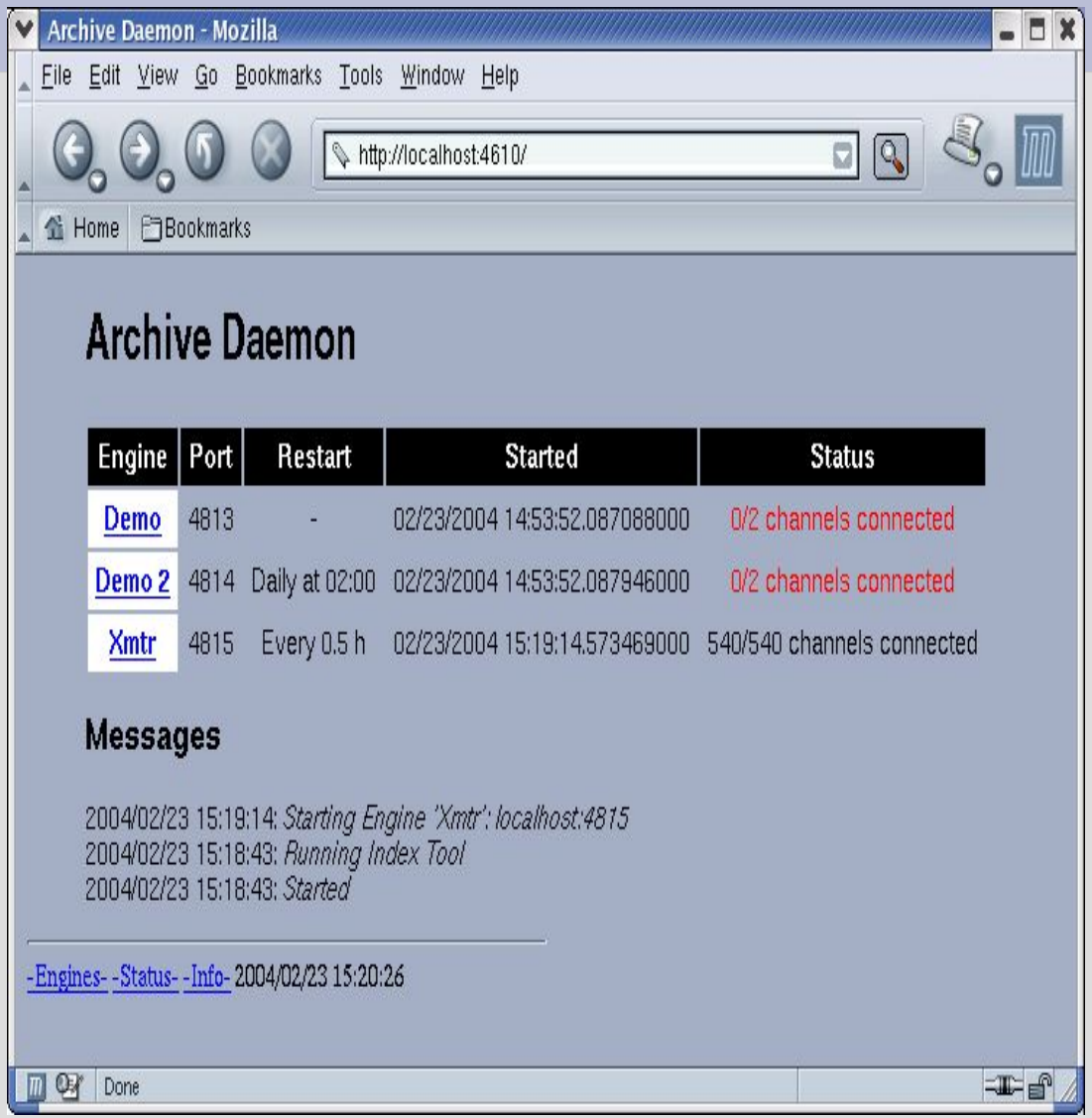
Performance: Retrieval



- ArchiveExport
  - single channel, raw
- ArchiveExport
  - 7 channel “filled” spreadsheet (95000 lines)
- Matlab-to-Dataserver
  - 7 ch. sheet (only 1st 500 lines, plus 1 sec. delay ML ... server ... ML)
- Matlab-to-Server
  - 7 ch into 500 “plot” bins (about 1500 vals/channel)

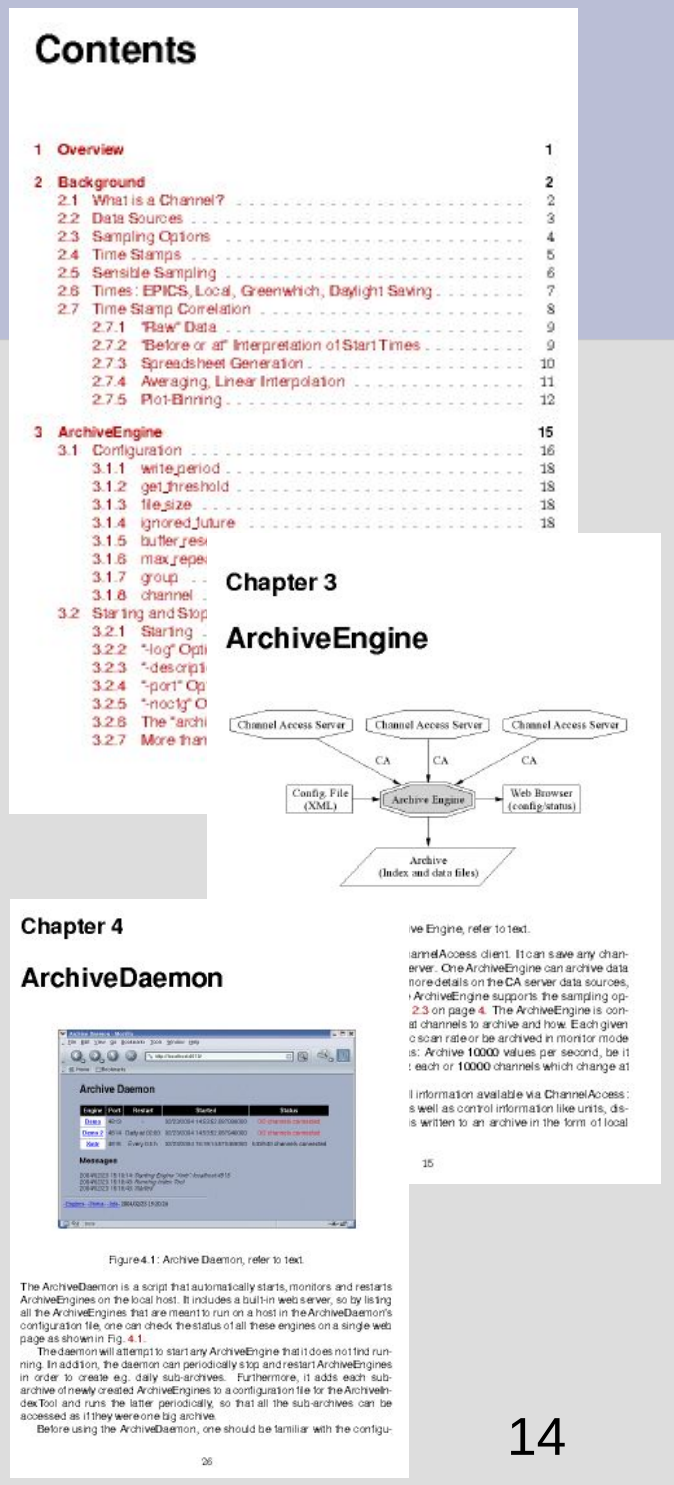
ArchiveDaemon

- Starts/restarts Engines similar to Thomas Birke's CabgManager (but perl, no cfg.-GUI).
- XML Config, HTTPD.
- Runs IndexTool to update master index for managed sub-archives.



Manual

- There is one, currently about 80 pages.
  - PDF, with index, printable, searchable, replacing the collection of web pages.
  - Details on what's a channel, record scan rates and sampling options in addition to description of archiver tool-set.



Contents

1 Overview 1

2 Background 2
2.1 What is a Channel? 2
2.2 Data Sources 3
2.3 Sampling Options 4
2.4 Time Stamps 5
2.5 Sensible Sampling 6
2.6 Times: EPICS, Local, Greenwhich, Daylight Saving 7
2.7 Time Stamp Correlation 8
2.7.1 "Raw" Data 9
2.7.2 "Before or at" Interpretation of Start Times 9
2.7.3 Spreadsheet Generation 10
2.7.4 Averaging, Linear Interpolation 11
2.7.5 Plot-Binning 12

3 ArchiveEngine 15
3.1 Configuration 16
3.1.1 write_period 18
3.1.2 get_threshold 18
3.1.3 file_size 18
3.1.4 ignored_future 18
3.1.5 buffer_res
3.1.6 max_repe
3.1.7 group
3.1.8 channel
3.2 Starting and Stop
3.2.1 Starting
3.2.2 "-log" Opti
3.2.3 "-descripti
3.2.4 "-port" Op
3.2.5 "-noctg" O
3.2.6 The "archi
3.2.7 More than

Chapter 3

ArchiveEngine

Chapter 4

ArchiveDaemon

Figure 4.1: Archive Daemon, refer to text.

The ArchiveDaemon is a script that automatically starts, monitors and restarts ArchiveEngines on the local host. It includes a built-in web server, so by listing all the ArchiveEngines that are meant to run on a host in the ArchiveDaemon's configuration file, one can check the status of all these engines on a single web page as shown in Fig. 4.1.

The daemon will attempt to start any ArchiveEngine that it does not find running. In addition, the daemon can periodically stop and restart ArchiveEngines in order to create e.g. daily sub-archives. Furthermore, it adds each sub-archive of newly created ArchiveEngines to a configuration file for the ArchiveIndexTool and runs the latter periodically, so that all the sub-archives can be accessed as if they were one big archive.

Before using the ArchiveDaemon, one should be familiar with the configu-

26

Current Participants, More...

- Sergei Chevtsov
- Kay-Uwe Kasemir
- Craig McChesney
- Peregrine McGehee
- Ernest Williams

- ChannelArchiver link on http://aps.anl.gov/epics.